

CLUBE

Z

80

/83

N.º 8

OLUBRE

2

[Faint horizontal line or bar at the bottom of the page]

AOS INTERESSADOS NA LANGUAGEY MAQUINA

Periodicamente chegam-nos pedidos de informacoes sobre o manerio de Linguagem Maquina, especialmente provenientes de utilizadores de maquinas da SINCLAIR.

Em portugal sempre se escreveu muito pouco sobre temas tecnicos. Nem sempre existe a coragem (e porque nao a incenuidade) de publicar textos sobre campos de accao em que os nacionais nao possuem tradicao de dominar.

Pessoalmente, conhecemos uma excecao - trata-se do eng. Moura Relvas - autor da "Introducao a Electronica Digital" e recentemente publicou "Introducao aos Microcomputadores".

Tivemos agora a excelente surpresa de ver as provas de aditamento (para a nova edicao) do livro "Introducao aos Microcomputadores" e que inclui cerca de uma centena de paginas dedicadas ao manerio do codigo maquina do Z-80, com applicacao no ZX 81 (directamente) e com algumas alteracoes podera eventualmente ser applicado tambem ao Spectrum.

Trata-se de uma excelente visao de todo o funcionamento do microcomputador, visto no sentido geral (maquinas de 8 bits) e com uma discussao do tipo passo a passo, no sentido de compreender como funcionam os registos internos do microprocessador e como se pode estruturar um programa em linguagem maquina.

Nao queremos criar ilusoes demasiado optimistas, no sentido de afirmar: - Lido o livro do Moura Relvas, toda a gente programa em linguagem maquina! Programar em linguagem maquina exige uma compreensao razoavel do funcionamento interno do microprocessador. Exige ainda muita disciplina e metodo de trabalho e uma boa dose de paciencia.

O que queremos deixar assinalado nestas linhas e a alegria de ver um texto claro, escrito por um homem que tem dedicacao a actividade de ensinar e divulgar conhecimentos. Uma boa parte da sua vida, e que nos temos a possibilidade de dissipar uma boa parte das duvidas (em relacao ao codigo maquina do Z-80), quando for lida a nova edicao da "INTRODUCAO AOS MICROCOMPUTADORES" que sera lancada estes dias nas livrarias.

 INTRODUCAO A
 LINGUAGEM
 MAQUINA

Z
 X
 -
 8
 1

=====

FERNANDO D'ALMEIDA PRECES

=====

Contin. do numero anterior

Continuemos entao...
 RUN 1000

Comentarios ao Programa 1
 a) A lentidao da execucao
 Verifica-se que o programa
 cumpre o especificado mas que em
 contrapartida e muito lento na
 sua execucao (cerca de 20
 segundos)

b) A estranha configuracao do
 Basic (rotina 1000).
 Muitos tecnicos de programacao
 chamam a este tipo de Basic um
 "Basic de aproximacao", ou ainda
 "pseudo-Basic", pois na
 realidade existe ja nesta rotina
 uma certa afinidade por
 aproximacao ao Assembler Z80

(conjunto de mnemonicas dadas
 pelo fabricante da
 maquina)-pag.187 do livro ZX.
 claro que isto foi intencional
 e, na minha opiniao, considero
 esta aproximacao uma etapa muito
 importante para todos os
 iniciados no codigo maquina.

(Todas as rotinas dum programa
 que se queiram passar a codigo
 maquina devem, em primeiro
 lugar, ser transformadas em
 "Pseudo Basic").

c) Interpretacao da rotina,
 linha a linha
 1000 - coloca HL na localizacao
 da memoria onde se encontra o
 primeiro caractere da mensagem.
 1015 - coloca em A o conteudo
 dessa localizacao

1020 - compara o conteudo de A
 com o caractere ** (codigo 216).
 1025 - se o resultado dessa
 comparacao for zero, salta para
 a linha 1050

1030 - manda imprimir o
 caractere contido em A
 1035 - incrementa HL para
 apontar a localizacao seguinte
 na memoria
 1040 - repeticao do ciclo.

SEGUNDA ETAPA - Elaboracao duma
 rotina em codigo maquina que
 substitua esta rotina Basic

cod.dec.	cod.hex.	mnemonicas
35	21	LD HL,+NN
148	94	} 16532
64	40	
(comentarios: aponta o regis- to par HL para a primeira localizacao)		
126	7E	LDA,(HL)
(com: coloca em A o conteudo de 16532)		
254	FE	CP, N
216	D8	**
(com: compara o registo A com este caractere)		
200	C8	RET Z
(com: retorno ao Basic, se der zero)		
215	D7	RS 16
(com: salto para a ROM (ro- tina. escreve um caractere)		
35	23	INC HL
(com: aponta HL para a locali- zacao seguinte)		
195	C3	JP, NN
133	85	} 16517
64	40	
(com: salto incondicional pa- ra esta localizacao)		
256		- Fim da codi- ficacao

TERCEIRA ETAPA - Introducao do
 codigo maquina na REM 1

```

ESCREVA:
1 REM XXXXXXXXXXXX (12
Caracteres)
120 PRINT,"3- CODIGO MAQUINA"
9700 LET X=16514
9705 PRINT,"INTRODUZA O CODIGO
EM DECIMAL"
9710 INPUT I
9715 IF I > 255 THEN RETURN
9720 POKE X,I
9730 PRINT AT 10,2: X,Y
9740 SCROLL
9750 LET X=X+1
9760 GOTO 9710
  
```

Depois de escrever este programa monitor que fara a introducao na REM 1, da rotina maquina, seleccione a rotina 3 e de entrada a sua codificacao.

QUARTA ETAPA - Introducao duma linha Basic que comande a rotina maquina.

Escreva:
500 RAND USR 16514
510 STOP

Grave primeiro o programa e depois
GOTO 500...

Comentarios ao programa 2

a) o programa cumpre o especificado
b) a execucao passou de 20 segundos para cerca de 1 segundo.

Outra vantagem:

A rotina em Basic ocupa 125 bytes da memoria.

A rotina maquina ocupa apenas 17 bytes.

Na continuacao do programa 2, podemos ainda efectuar mais uma experiencia, que se destina no fundamental a poupar espaco na RAM, quando da armazenagem de dados, mensagens ou tabelas que mais tarde queiramos reproduzir no ecran.

A rotina apresentada anteriormente em "pseudo-Basic" trabalha com linhas de 32 caracteres que, multiplicados pelas 22 linhas do ecran, representam 704 localizacoes ativas na RAM o que, para a maior parte dos casos, se torna num desperdicio de ocupacao da memoria.

Assim, no exemplo seguinte pretende-se utilizar a memoria, apenas para conter a informacao indispensavel, omitindo os espacos em branco em redor da mensagem.

Vejam os entao a apresentacao duma tabela da pontuacao dum determinado jogo, tendo em conta essa limitacao.

Chame a Rotina 2 e escreva:

```
JOGO+(ATAQUE+EM+TERRA)*PONTUACAO*
ASTEROIDES+...+10+PONTOS*PEQ.+NA
VES+...+20+PONTOS*GRAN.NAVES+...
.+50+PONTOS*DEPOSITOS++...+100PON
TOS**
```

Note que:

a) o simbolo (+) corresponde a um espaco em branco
b) o simbolo (*) e o caractere 23
c) o simbolo (**) e o caractere 216

Em seguida vamos alterar a Rotina em pseudo-Basic (linhas 1000 a 1055), substituindo o conteudo de algumas linhas e introduzindo outras.

Escreva:

```
1016 LET CP1=A-23
1017 IF NOT CP1 THEN GOTO 1045
1020 LET CP2=A-216
1025 IF NOT CP2 THEN GOTO 1070
1045 LET HL=HL+1
1050 LET A=118
1055 PRINT
1060 PRINT
1065 GOTO 1015
1070 STOP
```

Teste a actuacao desta rotina fazendo GOTO 1000 e se tudo estiver certo deve obter no ecran a tabela dos pontos como se exemplifica.

JOGO (ATAQUE EM TERRA)

Pontuacao

```
ASTEROIDES .... 10 PONTOS
PEQ.NAVES .... 20 PONTOS
GRAN.NAVES .... 50 PONTOS
DEPOSITOS ... 100 PONTOS
```

Agora reserve 30 caracteres para o codigo maquina escrevendo:

```
1 REM 1234567890 1234567890
1234567890
```

PROGRAMA EM CODIGO MAQUINA E COMENTARIOS

Loc.	Cod.	dex	Mnemon.	Pseudo-Basic
16514	33		LD HL,NN	LET HL=
	15	166		16550
	16	64		16550
	17	126	LD A,(HL)	-LET A=
				PEEK HL
	18	254	CP,N	LET CP1=
	19	23	(*)	A-23
	20	40	JZ,e	IF NOT CP1
	21	8	8	THEN GOTO
	22	254	CP,N	LET CP2=
	23	216	(**)	A-216
	24	40	JZ,e	IF NOT CP2
	25	12	12	THEN GOTO

Loc.	Cod.	dex.	Mnemon.	Pseudo-Basic
16526	215		RST 16	-(ROM-ROTINA ESCREVE CARACT.)
27	35		INC HL	-LET HL=HL+1
28	24		JR,e	} GOTO (256-243)
29	243		(256-243)	
30	35		INC HL	LET HL=HL+1
31	62		LDA,N	} LET A=118
32	117		117	
33	60		INC A	
34	215		RST 16	(ROM-ROTINA ESCREVA CARACTERE)
35	215		RST 16	} GOTO (256-235)
36	24		JR,e	
37	235		(256-235)	
38	201		RET	(RETORNO AO BASIC)

0000	- RST
007E	- TABUAS DE CARACTERES
0207	- PROJECCAO
02F6	- COMANDOS SAVE E LOAD
03CB	- INICIACAO
0419	- EDITORAS DA LINHA BASIC
0C29	- COMANDO DAS TABELAS BASIC
0CBA	- PESQUISA DA LINHA BASIC
0DAB	- COMANDO DO BASIC
0F52	- AVALIACAO DA EXPRESSAO BASIC
1586	- MANIPULACAO DO PONTO FLUTUANTE
1914	- TABUAS DAS FUNCOES
199C	- CALCULO DO PONTO FLUTUANTE
1E00	- GERADOR DE CARACTERES
1FFF	- Fim da ROM

Introduza estas instrucoes pelo codiso decimal, chamando a Rotina 3. Nao se esqueca de gravar o programa antes de ... GOTO 500, pois pode haver erros e perder tudo quanto ja fez.

Comentarios ao Programa 2a

- o programa cumpre o especificado
- o tempo de execucao continua a ser de aproximadamente 1 segundo.
- a rotina em pseudo-Basic ocupa localizacoes de memoria
- a mesma rotina em codigo maquina apenas ocupa 24.

Ate agora nos programas ja elaborados utilizamos unicamente uma rotina da ROM, a rotina RST 16 (ESCREVE UM CARACTERE); No entanto temos a nossa disposicao muitas outras que podemos utilizar na nossa programacao em codigo maquina, poupando assim muito tempo e muito espaco na memoria. Para tal teremos de conhecer o interior da ROM, o que iremos fazendo ao longo deste curso.

Como podem verificar, a rotina ja utilizada (a RST 16) faz parte do primeiro grupo, que contem as seguintes rotinas:

Primeiro grupo de rotinas

0000	ARRANQUE
0008	RELATORIO DE UTILIZACAO
0010	ESCREVE UM CARACTERE
0018	ENCAMINHA CARACTERE DUMA LINHA BASIC
0020	ENCAMINHA O PROXIMO
0028	SALTO PARA O CALCULADOR DE PONTO FLUTUANTE
0030	CRIA ESPACO NA MEMORIA
0038	INTERRUPCAO M1 POR CADA LINHA DE PROJECCAO
0066	INTERRUPCAO MN PARA A PROJECCAO EM SLOW

Repare que a localizacao das rotinas na ROM esta em numeracao hexadecimal.

Ainda no ambito do programa 2, vamos utilizar mais uma rotina pertencente agora ao grupo de projeccao (08F5 - rotina de teste dos parametros PRINT AT) muito importante para o trabalho em curso, por nos fazer aparecer as linhas da mensagem exactamente no ponto do ecran em que queiramos que elas aparecam. Para tal vamos alterar novamente a rotina em pseudo-Basic, como se segue:

As rotinas da ROM podem ser divididas em 14 grandes grupos, como se mostra a seguir.
ROM - Programa monitor de 8K bytes em linguagem maquina.

Eis os 14 grupos das rotinas:

CONTINUA NO
PROXIMO NUMERO

=====

SECCAO DO LEITOR

=====

SUGESTOES...COMENTARIOS...OPINIOES...DUVIDAS...SUGESTOES...COMENTARIOS...OPINIO

*"Sabendo que existem ligeiras
diferenças entre a programação
do ZX81 e do SPECTRUM
(abstrahindo das cores, som,
fast, slow) qual o modo de
"traduzir" um programa do ZX81
para o SPECTRUM?"

JOAO FORJAZ/
/Lisboa

Em principio, praticamente
tudo o que este programado em
BASIC para o ZX81 passa
directamente (atraves do
teclado) para o SPECTRUM.

A excepção e sempre feita
quando existe pelo a RAND USR
... por exemplo, ou PLOT, ou
UNPLOT.

Brevemente aparecera um
programa que passa
automaticamente numa cassette com
programas gravados para o ZX81,
directamente no SPECTRUM.

=====

*"O programa CONTAS CORRENTES,
publicado no boletim 5, nao
mostra as contas ja metidas que
nao tenham sofrido qualquer
alteracao. O programa e so
exemplo e assim nao funciona
correctamente? Ha alguma
modificacao necessaria a fazer
para o SPECTRUM? Interessava-me
saber algo mais sobre um
programa deste tipo mas com menu
mais completo."

FERNANDO JORGE/
/Lisboa

O programa e apenas
exemplificativo e, embora tenha
30 contas, nao tem a rotina para
listar as 30. Iremos tentar
publicar um pequeno programa que
de uma ajuda nesse sentido.

O CLUBE pode fornecer-lhe um
programa CONTAS CORRENTES
(SPECTRUM 48K), cujas instrucoes
lhe remetemos em separado.

=====

*"1 - O que e o modulo gerador
de som para o ZX81?"

2 - O que e o modulo gerador
de caracteres? Sera que este nos
permite construir o nosso
proprio conjunto de caracteres,
para alem dos ja existentes na
ROM do ZX81? Qual a resolucão
deste modulo e que zona de
memoria ocupa? Como programa-lo?

3 - Ainda nao ha em Portugal
modulos que permitam utilizar o
espaco de enderecamento que e
usado pelo eco da ROM, isto e,
entre 8192 e 16383? Se ha, onde
encontra-los (mesmo que so
permitem usar parte daquela
memoria)?

4 - Uma sugestao: Porque nao
inserir no boletim um espaco
destinado ao Hardware? Nao so
descriptivo mas tambem fornecendo
pequenos esquemas de construcões
electronicas para aumentar as
capacidades dos pequenos micros
como o ZX81."

MANUEL BORROES/
/Evora

1 - cremos que a pagina "os
efeitos, sonoros e o uso dos
microcomputadores" que incluimos
neste numero responde
parcialmente a sua questao.

2 - Nao possuímos elementos
suficientes para responder a
esta questao, pelo que pedimos a
colaboracao de outros leitores.
Ficamos desde ja a aguardar.

3 - Essa zona de memoria e
usada com codigo maquina
exclusivamente. Em Portugal
existem esses modulos: por
exemplo, o Interface Centronics
e o modulo de Alta Resolucao de
Graficos. E sempre possivel
usar, desde que o programa
esteja em codigo maquina.

4 - Tambem nos estamos de
acordo. O que e necessario e
colaboracao.

Quem possua alguma realizacao
ou tenha adaptado qualquer
esquema para o ZX81, podera
remeter para publicacao,

=====
SECCAO DO LEITOR
=====

Juntamente com os respectivos
comentarios.

=====

*"Responder-me que e
possivel fazer quaisquer
desenhos no ecran e que, para
isso se concretizar, posso
consultar o boletim 0. Como
posso eu por em pratica este
rotina? Ou seja, eu verificarei
que depois de a introduzir nada
se modifica e que os caracteres
continuam a ser os mesmos."

RUI CARVALHO/
/Barreiro

Relativamente ao desenho de
figuras no ecran, tem sido
publicadas algumas notas sobre
pequenas rotinas de uso geral;
no proximo numero saira um
programa muito completo,
adaptado da linguagem LOGO para
o ZX81, que o vai ajudar a
desenhar as figuras que quiser
no ecran. Nao se esqueca nunca
das limitacoes de resolucao do
ZX81.

=====

*"Deveriam mencionar nos
programas quais os "K" (...).
Inscricao para o curso Basic.
Pretendo trocar programas."

FERNANDO SOUSA/
/Apartado 150
2502 Caldas da RAINHA

Quanto ao primeiro aspecto,
apelamos aos leitores que
enviem programas que indiquem
os respectivos "K".

Relativamente ao curso, as
informacoes necessarias estao
incluidas no boletim 2.

=====

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
O LIVRO "COMPUTER PUZZLES" QUE
DIVULGAMOS NO NUMERO ANTERIOR
ESGOTOU.
COMO VARIOS LEITORES NOS
SOLICITARAM O SEU ENVIO,
FALE-MOS LOGO QUE RECEBAMOS
NOVO ORIGINAL.

=====
L I V R O S S P E C T R U M

BREVE PANORAMICA
=====

INTRODUCING SPECTRUM MACHINE
CODE

Ian Sinclair
Granada Publishing Ltd.
151 paginas

Trata-se fundamentalmente de
um manual de programacao
elementar, para o
microprocessador Z-80.

Constitui um bom ponto de
partida para principiantes,
requerendo, no entanto, outros
dados complementares no que diz
respeito a instrucoes mais
complexas.

Faz-se referencia ao programa
assembler ULTRAVIOLET, mas
deveriam ser fornecidas mais
informacoes quanto aos metodos
assembly "manuais", que sao
fundamentais para a compreensao
de todo este processo.

Estamos pois perante um
conjunto de informacoes gerais,
que exigem um aprofundamento
mais detalhado.

THE ZX SPECTRUM AND HOW TO
GET THE MOST FROM IT

Ian Sinclair
Granada Publishing Ltd.
130 paginas

Comecando por referir o modo
como sao feitas as conexoes
electricas e a ligacao ao TV,
este livro passa a exemplificar
principios de programacao
atraves de impressoes em
fac-simile.

Sao cerca de 70 exemplos que,
efectivamente, conseguem
ilustrar o texto, constituindo
uma boa ajuda para o
principiante.

20 BEST PROGRAMS FOR THE ZX SPECTRUM

Andrew Hewson
Hewson Consultants
118 paginas

Uma gama variada de programas, desde o Hangman a utilitarios em codigo maquina. As listagens sao apresentadas em facsimile da impressora Sinclair, sendo por vezes dificeis de decifrar, mas legiveis. Os utilitarios em codigo maquina sao provavelmente o mais importante dos programas.

PROGRAMMING YOUR ZX SPECTRUM

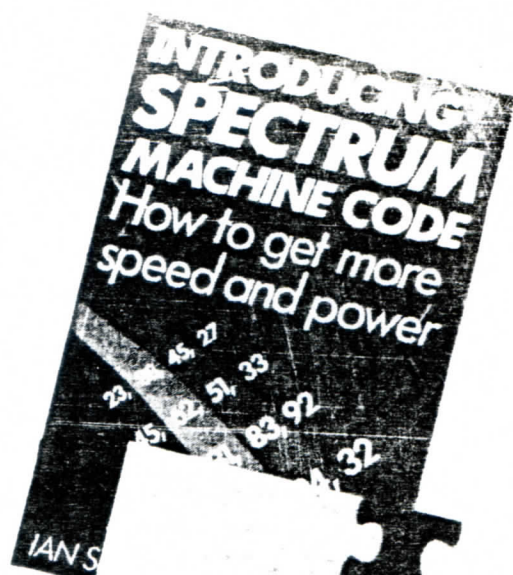
Tim Hartnell e Dilwyn Jones
Interface Publications
231 paginas

Um conjunto de programas bastante condensados, normalmente pequenos, seleccionados para ilustrar determinadas instrucoes e metodos.

As listagens sao em fac-simile e abrangem uma extensa zona. Algumas nao sao mais que adaptacoes do ZX-81 que, num ou noutro caso, sao perceptiveis. Um dos capitulos dedica-se a conversao de programas escritos para outros computadores.

A rapida sequencia de seccoes abordando diferentes temas origina, a primeira vista, uma certa confusao; contudo, o indice e suficientemente claro para permitir a procura rapida do tema desejado.

Enfim, nao e um manual muito aprofundado, mas contem material muito util para aqueles que pretendem compreender melhor o Basic.



IAN S

COMPUTER PUZZLES:
FOR SPECTRUM & ZX81

Ian Stewart and Robin Jones



IAN S

Programming
your
ZX SPECTRUM

=====

MAIS 2 LIVROS NO CLUBE Z-80

*GAMES TO PLAY ON YOUR ZX SPECTRUM
Martin Wren-Hilton
Preco (em fotocopias): 180\$00

*BETTER PROGRAMMING FOR YOUR SPECTRUM AND ZX81
S. Robert Speel
Preco (em fotocopias): 560\$00

=====

Programa
ALVO DESTRUIDO

F.Preces/
/Sacavem

```

0 REM PROGRAMA TRAD.E MODIF.
POR ALMEIDA PRECES, EM 3/4/82.
2 REM "3"
10 GOSUB 1000
12 LET T=0
14 LET X=PI+PI
15 LET U=1
20 LET Y=X+PI
30 LET Z=U
40 PRINT AT Y,X;"*"
45 LET T=T+1
50 SCROLL
60 LET X=X+(INKEY$="M" AND X<3
0)-(INKEY$="Z" AND X>PI)
62 LET U=U+Z
65 IF AND>.7 THEN GOTO 40
70 PRINT AT RND*15+5,RND*30;"*
"
80 PRINT AT RND*15+5,RND*30;"*
"
90 PRINT AT Y,X;
100 IF T=150 THEN GOTO 145
101 IF T=220 THEN GOTO 145
105 IF PEEK (PEEK 16398+256*PEE
K 16399)<>8 THEN GOTO 40
110 FOR O=0 TO 3
115 PRINT AT Y-2,X-3;"B A N G";
AT Y-1,X-1;" ";AT Y,X-1;" ";
AT Y+1,X-2;"B A N G";
120 PRINT AT Y-2,X-3;" ";
AT Y-1,X-1;" ";AT Y,X-1;" ";
AT Y+1,X-2;" ";
125 NEXT O
130 GOTO 278
145 PRINT AT Y,X;"0"
200 PRINT AT 15,10;"*
"
210 PRINT AT 18,10;"*
";AT 19,10;"*
"
215 LET F$=INKEY$
220 FOR B=0 TO 12
225 IF INKEY$<>"F" THEN GOTO 22
5
230 PRINT AT Y+1,X;"BANG"
235 PRINT AT Y+1,X;" "
240 PRINT AT Y+1,X;"BANG"
245 PRINT AT Y+1,X;" "
248 LET G=INT (RND*30)
249 IF G<8 OR G>25 THEN GOTO 24
8
250 PRINT AT 19,G;"*"
255 PRINT AT 19,G;" "
260 PRINT AT 19,G;"*
"
265 PRINT AT 19,G;" "
270 IF G=16 THEN GOTO 300
275 PRINT AT Y,X;" "
278 PRINT AT 5,3;"O SEU ROBOT F
OI DESTRUIDO"
280 GOTO 910
300 FOR O=0 TO 8
320 PRINT AT 5,7;"ALVO DESTRUID
O"
330 PRINT AT 19-0,16+0;"*
"
340 PRINT AT 19-0,16-0;"*
"
350 PRINT AT 5,7;"ALVO DESTRUID
O"
355 NEXT O
360 FOR O=0 TO 200
390 NEXT O
900 CLS
905 PRINT AT 8,0;"DESTRUICAO TO
TAL. PONTUACAO :";U
910 PRINT "QUER CONTINUAR? (D
IGA S OU N)."
920 INPUT O$
930 IF O$="S" THEN GOTO 980
940 PRINT "OK...OBRIGADO."
950 STOP
980 CLS
990 RUN 12
1000 PRINT AT 1,7;"ALVO DIFICIL"
1010 PRINT "O SEU ROBOT ES
TA ATRAVESSANDO"
1020 PRINT "UM CAMPO MINADO. E
O OBJECTIVO E"
1030 PRINT "DESTRUIR UMA IMPOR
TANTE CENTRAL."
1040 PRINT "VOCE VAI GUIA-LO
COM AS TECLAS:"

```

```

1050 PRINT "Z" PARA A ESQU
ERDA, "M" PARA A
1060 PRINT "DIREITA. COM "F"
FAZ FOGO QUANDO"
1065 PRINT "CHEGAR A CENTRAL."
1070 PRINT "PRIMA "N/L" PARA
COMECAR."
1080 INPUT O$
1090 CLS
1099 RETURN

```

Programa
MATE

Manuel Guinaz/
/Porto

```

5 REM "MATE"
10 RAND
20 SLOW
30 LET F=0
50 CLS
60 PRINT "ESCOLHA O TIPO DE OP
ERACAO "
62 PRINT "(1,2,3 OU 4)"
63 PRINT
64 PRINT "1) + SOMA"
65 PRINT "2) - SUBTRACAO"
66 PRINT "3) * MULTIPLICAO"
67 PRINT "4) / DIVISAO"
70 INPUT A
75 PRINT
80 PRINT "NIVEL 1,2 OU 3 ?"
90 INPUT B
100 FOR N=0 TO 10
110 CLS
120 PRINT "PERGUNTA NUMERO ";N,
F; " RESPOSTAS CERTAS"
125 IF N=10 THEN GOTO 278
130 LET C=INT (10*B+RND)
140 LET D=INT (10*B+RND)
150 IF A>2 THEN LET E=INT (10*(
B*(B-1))+1)
155 LET B$=STR$ C+CHR$ (A+20)+B
TR$ D
170 PRINT AT 10,1;B$;"=";
180 INPUT D
190 PRINT D
200 IF ABS (VAL B$-D)>0.21 THEN
GOTO 240
210 PRINT "CERTO;PRIMA ENTER"
220 LET F=F+1
230 GOTO 250
240 PRINT "ERRADO;PRIMA ENTER"
250 INPUT A$
260 NEXT N
271 PAUSE 150
272 CLS
275 PRINT AT 12,0;"A SUA PONTUA
CAO E ";F;" CERTAS "
276 PRINT "EM 10"
278 IF F>7 THEN PRINT
279 IF F>7 THEN PRINT "VOCE E U
M SABIO A MATEMATICA"
280 PAUSE 400
283 CLS
287 PRINT AT 12,0;"PARA JOGAR O
UTRA VEZ PRIMA"
288 PRINT "QUALQUER LETRA."
289 IF INKEY$="" THEN GOTO 235
290 RUN
300 SAVE "MATE"
310 RUN

```

ESCOLHA O TIPO DE OPERACAO
(1,2,3 OU 4)

- 1) + SOMA
- 2) - SUBTRACAO
- 3) * MULTIPLICAO
- 4) / DIVISAO

IRUN

PROGRAMA --- PLOT SKETCH ---

ESTE PROGRAMA FUNCIONA NUM ZX81 COM 1 K DE MEMORIA RAM
 PODE MOVER O CURSOR COM AS TECLAS 5,6,7,8. COM UM POUCO DE PRATICA, PODERA DESENHAR OS SEUS DESENHOS.

COM A TECLA (SHIFT 0) PODE ELIMINAR A PARTE DO TRACO QUE DESENHOU E QUE DEIXOU DE LHE INTERESSAR

PARA REGRESSAR AO MODO DE DESENHAR USE A TECLA DE (GRAPHICS)

OBS. SOBRE O PROGRAMA se quiser usar a instrucao UNPLOT em vez de PLOT substitua a linha 30 ... LET RS=0

A AREA DO DESENHO PODE SER AUMENTADA SE TEM MAIS DE 1 K DE MEMORIANA LINHA 80 ...LET X=X+(X<1)-(X>60).....90.....LET Y=Y+(Y<1)-(Y>40)

QUANDO QUISER PARAR use a tecla BREAK

1 REM PLOTSKETCH

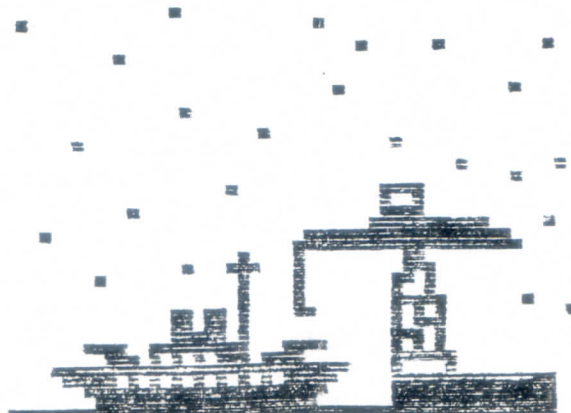
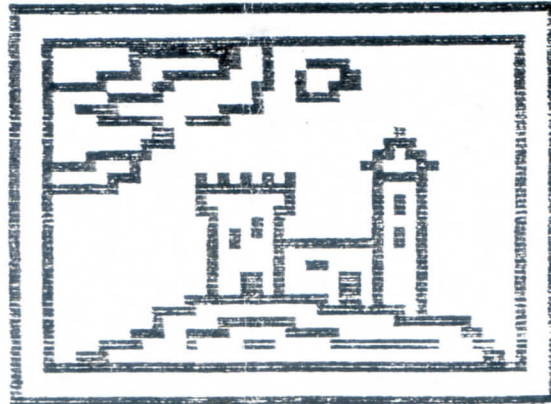
```

10 LET X=1
20 LET Y=5
30 LET RS=1

40 PLOT X,Y
50 IF RS<1 THEN UNPLOT X,Y
60 LET X=X+(INKEY$="8")-(INKEY$="5")
70 LET Y=Y+(INKEY$="7")-(INKEY$="6")
80 LET X=X+(X<1)-(X>50)
90 LET Y=Y+(Y<5)-(Y>40)

100 LET RS=SGN (RS+(INKEY$="9")-(INKEY$="0"))
110 GOTO 40

```




```

100 GO SUB 2000: GO SUB 5000: P
RINT "Prima uma tecla": PAUSE
0
110 GO SUB 6000
120 IF RND<=.5 THEN PRINT AT 21
0; PAPER 2; FLASH 1; "E VOCE A J
OGAR " : PAUSE 1
00: GO TO 190
130 PRINT AT 21,0; PAPER 1; FLA
SH 1; "SPECTRUM A JOGAR
" : PAUSE 100
140 PRINT AT 21,0; PAPER 1; FLA
SH 1; "SPECTRUM JOGANDO
" : LET d=1: LET e=a(d): GO
SUB 7000: LET a=1+INT (6*RND):
IF NOT c AND a(a) THEN LET b=1:
GO TO 180
150 IF NOT c THEN GO TO 140
160 FOR d=1 TO 6: FOR e=a(d)-1
TO 0 STEP -1: GO SUB 7000: IF c
THEN NEXT e: NEXT d
170 LET a=d: LET b=a(a)-e
180 GO SUB 8000: GO SUB 9000: L
ET a=0: FOR n=1 TO 6: LET a=a+(
n): NEXT n: IF a=0 THEN PRINT AT
21,0; PAPER 1; FLASH 1; "SPECTRU
M esta a ganhar " : INPUT
I "Prima outra vez uma tecla": L
INE z$: GO TO 110
190 PRINT AT 21,0; PAPER 2; FLA
SH 1; "Agora jogus voce
" : INPUT AT 0,0; INK 2; PA
PER 7; "Com qual voce ataca ?Indi
que um numero seguido por ENTER,
" : LINE z$: IF z$="" THEN IF (
LEN z#=1) AND (z#="0") AND (z#<=
"6") THEN LET a=VAL z$: IF a(a)
0 THEN GO SUB 8000: GO TO 210
200 GO TO 190
210 INPUT AT 0,0; PAPER 2; "Quan
tos voce remove ?Indique um nume
ro seguido por ENTER. " : LINE z$:
IF z$="" THEN IF (LEN z#=1) AN
D (z#="0") AND (z#<="7") THEN LE
T b=VAL z$: IF b(=a(a) THEN GO T
O 230
220 GO TO 210
230 GO SUB 9000: LET a=0: FOR n
=1 TO 6: LET a=a+(a): NEXT n: I
F a=0 THEN PRINT AT 21,0; PAPER
2; FLASH 1; "VOCE GANHOU
" : INPUT AT 0,0; "Pri
ma ENTER para outro jogo": LINE
z$: GO TO 110
240 GO TO 140
2000 REM texto
2010 INK 7: PAPER 0; BORDER 0: C
LS : PRINT PAPER 2; "

```

ANDROID NIM

" PAPER 1;
" Nim e um jogo a dois, o seu
" parceiro e o spectrum.

2015 PRINT " PAPER 1;" Escolha
a fila, removendo
qualque
r numero de androids
dessa r

2020 PRINT " PAPER 6; INK 0;"
O vencedor e o que remove o

Ultimo Android

```

2030 RETURN
5000 DATA 7,31,21,17,17,15,7,1
5010 DATA 7,31,17,25,17,15,7,1
5020 DATA 7,31,17,19,17,15,7,1
5030 DATA 7,31,17,17,21,15,7,1
5040 DATA 224,248,168,136,136,24
0,224,128
5050 DATA 224,248,136,200,136,24
0,224,128
5060 DATA 224,248,136,152,136,24
0,224,128

```

```

5070 DATA 224,248,136,136,168,24
0,224,128
5080 DATA 31,19,31,3,3,1,1,3
5090 DATA 15,11,27,19,19,2,6,12
5100 DATA 255,131,3,3,31,16,48,0
5110 DATA 248,200,248,192,192,12
8,128,192
5120 DATA 240,208,216,200,200,64
96,48
5130 DATA 255,193,192,192,248,8,
12,0
5140 FOR n=0 TO 13: RESTORE 5000
+10*n: FOR m=USR CHR$(97+n) TO
USR CHR$(97+n)+7: READ o: POKE
m,0: NEXT m: NEXT n
5150 DIM a$(4,2): DIM b$(3,2): L
ET a$(1)="0000": LET a$(2)="0000": L
ET a$(3)="0000": LET a$(4)="0000": L
ET b$(1)="0000": LET b$(2)="0000": L
ET b$(3)="0000"
5160 RETURN
6000 REM sub to set up board
6010 CLS : PRINT AT 0,0; PAPER 1
: ANDROID NIM
" : DIM a(6): FOR n=1 TO 6: LET
a(n)=1+INT (RND*7): FOR m=1 TO a
(n): PRINT AT 3*n,3*m; INK 4+INT
(RND*3);a$(1);AT 3*n+1,3*m; I
N 3;b$(1);AT 3*n,0; INK 7;n: NEXT
m: NEXT n
6020 RETURN
7000 REM sub to determine safe o
r unsafe (c=0)
7010 DIM b(6,3)
7020 FOR n=1 TO 6: LET a=a(n): I
F n=d THEN LET a=e
7030 LET b(n,3)=INT (a/4): LET a
=a-4*INT (a/4): LET b(n,2)=INT (
a/2): LET a=a-2*INT (a/2): LET b
(n,1)=INT a: NEXT n
7040 DIM c(3): FOR n=1 TO 3: LET
c(n)=0: FOR m=1 TO 6: LET c(n)=
(c(n)+b(m,n)): NEXT m: NEXT n
7050 LET c=c(1)+c(2)+c(3): RETUR
N
8000 REM sub to point android in
right direction
8010 PRINT AT 3*a,0; FLASH 1;a
8020 FOR n=1 TO a-1: FOR m=1 TO
a(n): LET s=INT (3*RND): FOR p=2
TO 1 STEP -1: FOR q=3 TO 4: PRI
NT AT 3*n,3*m; INK 4+s;a$(q);AT
3*n+1,3*m; INK 3;b$(p): BEEP .02
,2*(q+m+n+p): NEXT q: NEXT p: NE
XT m: NEXT n
8030 FOR m=1 TO a(a): LET s=IN
T (RND*3): FOR p=1 TO 3: FOR q=2 T
O 3: PRINT AT 3*a,3*m; INK 4+s;a
$(q);AT 3*a+1,3*m; INK 3;b$(p):
BEEP .02,10+2*(q+p+m): NEXT q: N
EXT p: NEXT m
8040 FOR n=a+1 TO 6: FOR m=1 TO
a(n): LET s=INT (RND*3): FOR p=1
TO 2: FOR q=2 TO 1 STEP -1: PRI
NT AT 3*n,3*m; INK 4+s;a$(q);AT
3*n+1,3*m; INK 3;b$(p): BEEP .02
,20+2*(q+p+m+n): NEXT q: NEXT p:
NEXT m: NEXT n
8050 RETURN
9000 REM sub to remove android a
=row, b=no. to be removed
9010 FOR n=a(a) TO a(a)-b+1 STEP
-1: FOR p=2 TO 3: FOR q=1 TO 3:
FOR r=1 TO 4: PRINT AT 3*a,3*n;
INK 7;a$(r);AT 3*a+1,3*n; INK 2
;b$(q): BEEP .1-.03*p,40+r+p+q:
NEXT r: NEXT q: NEXT p
9020 PRINT AT 3*a,3*n; INK 5; FL
ASH 1;a$(4);AT 3*a+1,3*n; INK 2;
FLASH 1;b$(3): BEEP 1,60: PRINT
AT 3*a,3*n; " : AT 3*a+1,3*n; "
" : NEXT n
9030 LET a(a)=a(a)-b
9040 PRINT AT 3*a,0;a: RETURN

```



```

100 PAPER 0: INK 7: BRIGHT 1: B
ORDER 0: CLS: PRINT AT 0,10: PA
PER 1: " OHELLO "
110 GO SUB 1000
120 GO SUB 4000
130 IF AND(.5 THEN GO TO 170
140 PRINT AT 21,0: PAPER 2: " SP
ECTRUM A JOGAR "
150 GO SUB 3000
160 IF f1=2 THEN GO TO 220
170 PRINT AT 21,0: PAPER 2: " V
OICE A JOGAR "
180 LET c$=" "
190 GO SUB 5000
200 IF f1=2 THEN GO TO 220
210 GO TO 140
220 GO SUB 6000: PRINT AT 21,0:
PAPER 2: "SPECTRUM ";a: " VOICE
";v
230 INPUT "PRIMA ENTER PARA UM
NOVO JOGO : LINE z$: RUN
1000 REM set up
1010 DATA 0,52,4,12,13,5,53,1,59
,60,36,44,45,37,61,54,11,43,20,2
4,25,21,38,6,19,51,31,32,33,26,4
6,14,18,50,30,34,35,27,47,15,10,
42,23,29,28,22,39,7,58,63,41,49,
43,40,62,55,3,57,9,17,16,8,56,20
1020 DIM x(64): DIM y(64): FOR y
=1 TO 8: FOR x=1 TO 8
1030 READ a: LET x(a+1)=x: LET y
(a+1)=y
1040 NEXT x: NEXT y
1050 DATA 0,0,0,3,7,15,31,31,0,0
,0,192,224,240,248,248,31,31,15,
7,3,0,0,0,248,248,240,192,0,
0,0
1060 DATA 1,1,49,25,13,7,3,1,128
,128,140,152,176,224,192,128,0,0
,32,48,24,12,6,255,255,6,12,24,4
8,32,0,0
1070 FOR n=0 TO 7: FOR m=0 TO 7:
READ a: POKE USR CHR$(n+CODE
"+m),a: NEXT m: NEXT n
1080 DIM p$(2,2): DIM a$(2,2)
1090 FOR j=1 TO 2: FOR k=1 TO 2:
LET p$(1,j,k)=CHR$(2*j+k+141):
LET a$(j,k)=CHR$(2*j+k+145): N
EXT k: NEXT j
1100 DIM b$(8,8): LET b$(5,5)="
": LET b$(4,4)="": LET b$(5,4)=
": LET b$(4,5)=" "
1110 RETURN
1500 REM evaluate move
1510 LET f1=0
1520 LET d$="": IF c$=d$ THEN L
ET d$=" "
1530 IF b$(y1,x1)<>" " THEN RETU
RN
1540 FOR y=-1 TO 1: FOR x=-1 TO
1
1550 LET y2=y1+y: LET x2=x1+x
1560 IF y2=0 OR y2=9 OR x2=0 OR
x2=9 OR NOT (x OR y) THEN GO TO
1590
1570 IF b$(y2,x2)=d$ THEN GO SUB
2000
1580 IF f1 THEN RETURN
1590 NEXT x: NEXT y
1600 RETURN
2000 REM eval move direction
2010 LET y3=y2: LET x3=x2
2020 LET y3=y3+y: LET x3=x3+x: I
F y3=0 OR y3=9 OR x3=0 OR x3=9 T
HEN RETURN
2030 IF b$(y3,x3)=d$ THEN GO TO
2040
2040 IF b$(y3,x3)=" " THEN RETUR
N
2050 LET f1=1: RETURN
2500 REM make move

```

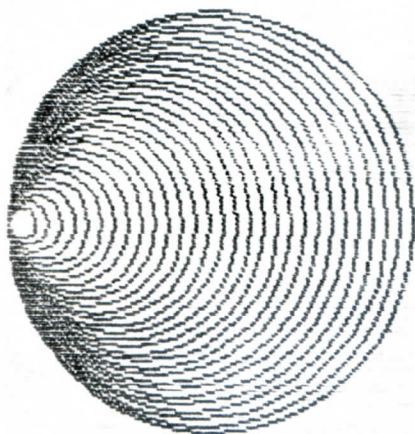
```

2510 LET d$="": IF d$=c$ THEN L
ET d$=" "
2520 LET b$(y1,x1)=c$: LET x3=x1
: LET y3=y1: GO SUB 4500
2530 FOR y=-1 TO 1: FOR x=-1 TO
1
2540 LET y2=y1+y: LET x2=x1+x
2550 IF y2=0 OR y2=9 OR x2=0 OR
x2=9 OR NOT (x OR y) THEN GO TO
2610
2560 LET f1=0: IF b$(y2,x2)=d$ T
HEN GO TO 2000
2570 IF NOT f1 THEN GO TO 2610
2580 LET b$(y2,x2)=c$: LET y3=y2
: LET x3=x2: GO SUB 4500: LET y2
=y2+y: LET x2=x2+x
2590 IF b$(y2,x2)=c$ THEN GO TO
2610
2600 GO TO 2530
2610 NEXT x: NEXT y
2620 RETURN
3000 REM computer's move
3010 LET c$="": PRINT AT 2,6: "
PRINT AT k,4: " ": FOR k=4 TO 19:
3020 FOR k=1 TO 64: BEEP .1,k: L
ET y1=y(k): LET x1=x(k): GO SUB
1500
3030 IF NOT f1 THEN NEXT k
3040 IF f1 THEN GO SUB 2500: RET
URN
3050 LET f1=2: RETURN
4000 REM print board
4010 LET f1=1
4020 FOR y=1 TO 8: FOR x=1 TO 8
4030 LET y3=y: LET x3=x: GO SUB
4500
4040 NEXT x: NEXT y
4050 LET f1=0: RETURN
4500 REM print piece
4510 FOR j=1 TO 2
4520 LET p=1+(b$(y3,x3)=" "): PR
INT AT 1+2*y3+j,4+2*x3: INK 1+(b
$(y3,x3)=""): PAPER 7-((x3+y3)/
2=INT((x3+y3)/2)):P$(p,j)
4530 NEXT j
4540 IF f1=0 THEN FOR k=30 TO 15
STEP -1: BEEP .008,k: NEXT k
4550 RETURN
5000 REM user's move
5010 LET s=1: LET r=1
5020 GO SUB 5500
5030 LET z$=INKEY$: IF z$="" THE
N GO TO 5030
5040 BEEP .1,20: LET s=s+(z$="8"
AND s<8)-(z$="5" AND s>1)
5050 LET r=r+(z$="6" AND r<8)-(z
$="7" AND r>1)
5060 IF z$="0" THEN LET x1=s: LE
T y1=r: GO SUB 1500: IF f1 THEN
GO SUB 2500: RETURN
5070 IF z$="1" THEN LET f1=2: RE
TURN
5080 GO TO 5020
5500 REM print arrows
5510 PRINT AT 2,6: "
5520 FOR j=4 TO 19: PRINT AT j,4
: " ": NEXT j
5530 PRINT AT 2,4+2*s;3$(1)
5540 FOR j=1 TO 2: PRINT AT 1+2*
r+j,4;a$(2,j): NEXT j
5550 RETURN
6000 REM score
6010 LET a=0: LET b=0
6020 FOR y=1 TO 8: FOR x=1 TO 8
6030 LET a=a+(b$(y,x)=""): LET
b=b+(b$(y,x)=" ")
6040 NEXT x: NEXT y
6050 RETURN

```


Graficos e som no seu SPECTRUM

Os dois programas foram feitos por Francisco Santos de Setubal

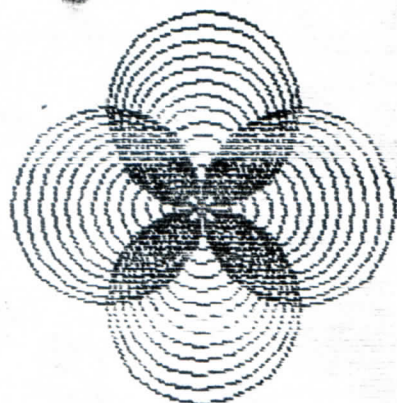


```

1 REM "A concha"
10 REM Feito por Francisco Santos
20 FOR g=150 TO 10 STEP -5: BE
EP .05,3
30 PLOT 0,68: DRAW g,0,PI: BEE
P .05,4: DRAW -g,0,PI
40 NEXT g

```

Fernando Santos/
Setubal



```

1 REM Flor
10 REM Feito por Francisco Santos
20 FOR g=10 TO 70 STEP 5
30 PLOT 126,68: DRAW -g,0,PI:
BEEP .03,0: DRAW g,0,PI
40 BEEP .03,2: DRAW g,0,PI: B
EP .03,1: DRAW -g,0,PI
50 BEEP .03,3: DRAW 0,g,PI: BE
EP .03,0: DRAW 0,-g,PI
60 BEEP .03,2: DRAW 0,-g,PI: B
EEP .05,1: DRAW 0,g,PI: BEEP .03
,3
70 NEXT g

```

```

1 REM
-RELOGIO-
Adaptado por Francisco Santos
110 INPUT "Horas",e
120 IF e>24 THEN GO TO 110
130 IF e>12 THEN LET e=e-12
140 INPUT "Minutos",m
150 IF m>59 THEN GO TO 140
160 LET d=INT (m/10): LET c=m-d
*10
170 INPUT "Segundos",s
180 IF s>59 THEN GO TO 170
190 LET b=INT (s/10): LET a=s-b
*10
300 PRINT AT 11,12;e;":":d;c;":
":b;a;
310 LET a=a+1
320 IF a>=10 THEN LET b=b+1: LE
T a=0
330 IF b>=6 AND a=0 THEN LET c=
c+1: LET b=0
340 IF c>=10 THEN LET d=d+1: LET
c=0
350 IF d>=6 AND c=0 THEN LET e=
e+1: LET d=0
360 IF e>=13 THEN LET e=1
370 PAUSE 46,5
380 GO TO 300

```

```

2500 REM "Circulo"
2504 INPUT "x-":x
2505 INPUT "y+":y
2510 INPUT "raio":r
2530 GO TO 2580
2550 LET l=INT ((SQR ((r+2)-(c+2
))) +.5)
2570 RETURN
2620 FOR c=r-1 TO 0 STEP -1
2630 GO SUB 2550
2635 PLOT (x-l),(y+c)
2645 DRAW (2+l),0
2650 NEXT c
2660 FOR c=1 TO r-1
2665 GO SUB 2550
2670 PLOT (x-l),(y-c)
2690 DRAW (2+l),0
2700 NEXT c
2710 GO TO 2504

```

Alteracoes

```

60 INK 1: PAPER 5: BORDER 5: C
LS
70 LET x=128: LET y=92: LET r=
50
80 GO SUB 2530
90 PAPER 7: INK 0
2710 RETURN

```

Os programas RELOGIO e CIRCULO podem ser usados em conjunto. Para isso deverao ser includidas as instrucoes constantes na lista de alteracoes. A juncao da um melhor efeito visual. Uma sugestao - caso queira comecar o programa com 0:00:00, introduzir a seguinte linha:
100 LET a=0: LET b=0: LET c=0:
LET d=0: LET e=0: GO TO 300

S P E C T R U M

Programa ESCAPE

J. MAGALHAES/
/Porto

JOGO N-1 PLASTERS=20

10 - caracteres do robot

15 - define caracteres do bloco

30-70 - cria os caracteres para
o robot e bloco110-210 - introducao e
instrucoes

220-6 - calculo de blocos

F - calculo de "plasters"

230-L - indicador do movimento
do robot

```

1 CLS : REM "escape"
2 PRINT "Programa para SINCLA
IR SPECTRUM 15 ou 48 k"

5 POKE 23658,8: REM caps lock
actuando
10 DATA BIN 00011000,BIN 00111
100,BIN 01100110,BIN 11000011,BI
N 01111110,BIN 00111100,BIN 1110
0111,BIN 01100110
15 LET a=BIN 01111110
20 DATA 0,a,a,a,a,a,a,0
30 FOR b=146 TO 147
40 FOR c=0 TO 7
50 READ d: POKE USR CHR$ b+c,d
60 NEXT c
70 NEXT b
100 REM "escape"
110 FLASH 1: PRINT AT 10,6;"Des
ligar o gravador": FLASH 0: PAUS
E 200
120 CLS : PRINT "O OBJECTIVO DE
STE JOGO E CONDUZIR O ROB
OT ATE AO PONTO- H"
130 PRINT AT 4,4;"PARA MOVER O
ROBOT USE AS TECLAS 5,6,7,
8"
140 PRINT AT 8,0;"INFELIZMENTE
O ROBOT TEM POUCA PROTECCAO CON
TRA OS BLOCOS <plasters> QU
E IMPEDEM OS SEUS MOVIMENTOS"
150 PRINT AT 15,10;"IMPORTANTE"
160 PRINT AT 17,0;"O ROBOT APEN
AS SOBREVIVE A 20 IMPACTOS,AP0
3 OS QUAIS FICA INUTILIZADO"
170 PRINT AT 21,0;"*Qualquer te
cla para continuar*"
180 PAUSE 0
190 CLS
200 PRINT AT 10,1;"se e impossi
vel para o robot chegar a casa
(H), entao use a tecla 1 e abo
rtara o jogo"
210 PRINT AT 10,7;"use qualquer
tecla"
220 PAUSE 0: LET E=3: LET F=20
230 LET L=0
240 BORDER 6: PAPER 6: INK 0: C
LS

```

```

250 LET A=20: LET B=30
260 PRINT AT 21,2;"JOGO N-";E-2
270 PRINT AT 21,16;"PLASTERS=";
F
280 PRINT AT 0,0;"H"
290 PRINT AT A,B;"A": REM GRAPH
IC C
300 FOR D=1 TO E
310 PAPER 2
320 PRINT AT INT (RND*21),INT (
RND*32);"■": REM GRAPIC D
330 NEXT D
340 PAPER 6
350 PRINT AT INT (RND*3),INT (R
ND*3);" "
360 IF INKEY$="1" THEN GO TO 71
370 IF INKEY$="5" THEN PRINT AT
A,B;" " : LET B=B-1: LET L=1
380 IF INKEY$="6" THEN PRINT AT
A,B;" " : LET A=A+1: LET L=2
390 IF INKEY$="7" THEN PRINT AT
A,B;" " : LET A=A-1: LET L=3
400 IF INKEY$="8" THEN PRINT AT
A,B;" " : LET B=B+1: LET L=4
410 IF A<0 THEN LET A=0
420 IF A>20 THEN LET A=20
430 IF B<0 THEN LET B=0
440 IF B>31 THEN LET B=31
450 IF ATTR (A,B)=16 THEN GO SU
B 480
460 IF A=0 AND B=0 THEN GO TO 5
50
470 GO TO 260
480 LET F=F-1
490 IF L=1 THEN LET B=B+1
500 IF L=2 THEN LET A=A-1
510 IF L=3 THEN LET A=A+1
520 IF L=4 THEN LET B=B-1
530 IF F=0 THEN PRINT AT 21,25;
"0": PRINT AT A,B;"A": PRINT AT
10,13;"FIM": PAUSE 200: GO TO 72
540 RETURN
550 BORDER 5: INK 1: PAPER 5: C
L S
560 FOR B=1 TO 5
570 INK 5
580 FOR A=0 TO 1
590 PRINT AT 10,10;"CORRECTO"
600 PAUSE 10
610 INK 1
620 NEXT A
630 NEXT B
640 INK 1
650 CLS
660 PRINT AT 12,7;"CONSEGUE REP
ETIR ?"
670 PRINT AT 14,7;"USE A TECLA
SPACE"
680 IF INKEY$=" " THEN GO TO 69
685 GO TO 680
690 LET E=E+1
700 GO TO 240
710 PRINT AT 10,12;"ABORTOU"
720 PRINT AT 12,6;"USE A TECLA
SPACE"
725 PRINT AT 14,4;"TECLA 'N' PA
RA TERMINAR"
730 IF INKEY$=" " THEN CLS : PA
USE 50: GO TO 120
735 IF INKEY$="N" THEN GO TO 10
00
740 GO TO 730
750 GO TO 240
1000 CLS : PRINT AT 10,0;"QUALQU
ER TECLA - LIMPAR MEMORIA": PAUS
E 0: NEW
1001 SAVE "ESCAPE" LINE 1

```

250 - posicao inicial do robot

260-270 - scores

280 - posicao de H (ponto de chegada)

290 - desenho do robot (grafico "c")

300-330 - localizacao dos blocos (ao acaso)

320 - grafico "d"

360-400 - controle de movimentos do robot

410-440 - mantem o grafico do robot no ecran

450 - colisao do robot

460 - quando o robot atinge o ponto de chegada "H"

470 - inicio de novo jogo

480-540 - fim do jogo, quando "plasters"=0

536 - grafico "c"

550-630 - "correcto". Quando o robot atinge o objectivo

680 - tecla para iniciar o jogo

710-100 - quando usar a tecla "1" (abortar o jogo)

NOTA - SE UTILIZAR A TECLA N, O PROGRAMA SERA APAGADO DA MEMORIA

SINCLAIR ZX 81

Com este programa inserido no final do seu programa, poderá em qualquer altura, saber qual o estado da memória, em termos de qual o espaço ocupado pelo programa, quer o espaço ocupado pelas variáveis usadas. Bastará para tanto, fazer BREAK e depois GOTO 9996

```
9995~STOP
9996 PRINT "ZX81-MEMORIA TOTAL=";(PEEK 16388+256*PEEK 16389)-16384)/1024;"K"
9997 PRINT ,,"PROGRAMA EM BASIC";(PEEK 16396+256*PEEK 16397)-16509;" BYTES"
9998 PRINT ,,"VARIABLES ";(PEEK 16404+256*PEEK 16405)-(PEEK 16400+256*PEEK 16401)-1;" BYTES"
9999 PRINT ,,"MEMORIA LIVRE";(PEEK 16388+256*PEEK 16389)-(PEEK 16404+256*PEEK 16405);" BYTES"
```

```
=====
Q U A L   O   E S T A D O   D A   M E M O R I A   ?
=====
```

SPECTEM

Com este programa inserido no final do seu programa, poderá em qualquer altura, saber o estado da memória, em termos de saber o espaço ocupado pelo programa, qual a capacidade da memória da sua máquina, também obtém o espaço ocupado pelas variáveis e ainda tem um relógio que começa a funcionar logo que dirija o programa para a linha 9997 (CAPS e BREAK) e depois GOTO 9997

```
9997 DEF FN p(n)=PEEK n+256*PEEK
(n+1)
9998 CLS : DRAW 255,0: DRAW 0,17
S: DRAW -255,0: DRAW 0,-175: PLO
T 40,158: DRAW 127,0: PRINT AT 1
,5;"Status da Máquina..";AT 4,2;
"Total MEMORIA RAM : ";(FN p(237
32)-16383)/1024;" K";AT 6,2;"Míc
rodrive Maps: ";FN p(23631)-23734
;" bytes";AT 7,2;AT 8,2;"Program
a Basic : ";FN p(23627)-FN p(2363
5);" bytes";AT 9,2;"Variáveis
";FN p(23641)-FN p(23627)-1
;" bytes";AT 11,2;"Espaco Livre
";FN p(23730)-FN p(23641);"
bytes";AT 19,2;"UDG:ABCDEFGHIJKL
MNOPQRSTU"
9999 PRINT AT 14,2;" Tempo actua
l ";INT ((FN p(23672)+65536*PE
EK 23674)/50);" segundos": GO TO
9999
```

 INTRODUÇÃO A
 PROGRAMAÇÃO
 ESTRUTURADA

 (cont. do número anterior)
 Última parte

Voltemos as arborescências.

É possível, conforme já referimos, transformar uma arborescência numa árvore binária fazendo uma alteração "canônica" (que funciona em todos os casos).

Esta transformação consiste em colocar a fila esquerda em direcção ao primeiro dos descendentes, e a fila direita em direcção aos elementos que estão ao mesmo nível da raiz. Assim, a linha direita transforma-se no "irmão mais novo". Veja-se a arborescência da fig. 9 (número anterior, pag. 22) que se transforma numa árvore binária - fig. 13. Os percursos "pre-ordem" e "post-ordem" tem efeito idêntico ao de uma árvore binária genuína; O efeito do percurso "in-ordem" consiste no tratamento de todas as filas, da raiz e dos "irmãos".

Ha outras possibilidades de transformar arborescências em árvores binárias. A figura 14 apresenta uma outra formulação da árvore zoológica.

Inúmeras estruturas mais complexas podem reduzir-se a uma árvore binária. As listas List, por exemplo, são implantadas desta maneira. É também possível representar um gráfico com o auxílio de uma árvore binária. O da figura 15 foi convertido neste forma. Foi literalmente "dividido" e redefinido como a sequência dos arcos procedentes das ligações do gráfico. Teriam sido possíveis outras representações. Com efeito é necessário frequentemente fazer uma selecção quanto ao modo de representação, que tenha em conta as necessidades, a eficácia de implantação e as possibilidades da linguagem de programação a que se dispõe.

lista-cabeça

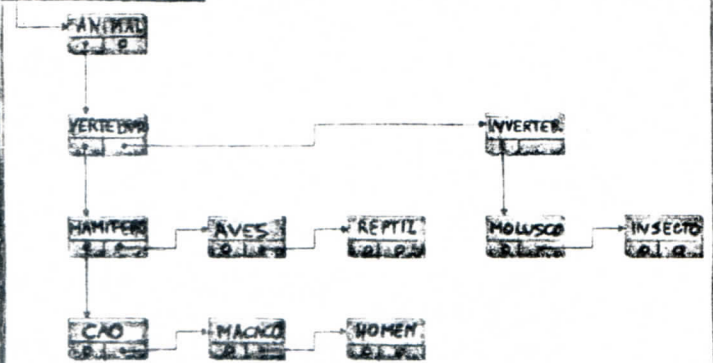


Fig. 13 - Representação canônica da arborescência da figura 17, em forma de uma árvore binária.

Acabamos assim de passar em revista algumas das estruturas de dados mais clássicas em informática. É evidente que esta lista não é exaustiva; estruturas mais complexas aparecem sem cessar, sobretudo no campo da Inteligência Artificial, mas a sua extensão acaba por ser idêntica à das estruturas aqui mencionadas.

Acentuamos principalmente a diferença entre a utilização lógica de uma estrutura de dado e a sua representação física. Esta distinção é muito importante pois permite trabalhar com estruturas lógicas durante a fase de concepção lógica, em que as escolhas de representação ainda não foram efectuadas. De certo modo, pode dizer-se que em informática é mais importante o "conceito" que a sua implementação. Os grandes progressos destes últimos anos devem-se mais a uma ideia nova que a um novo modo de representação.*

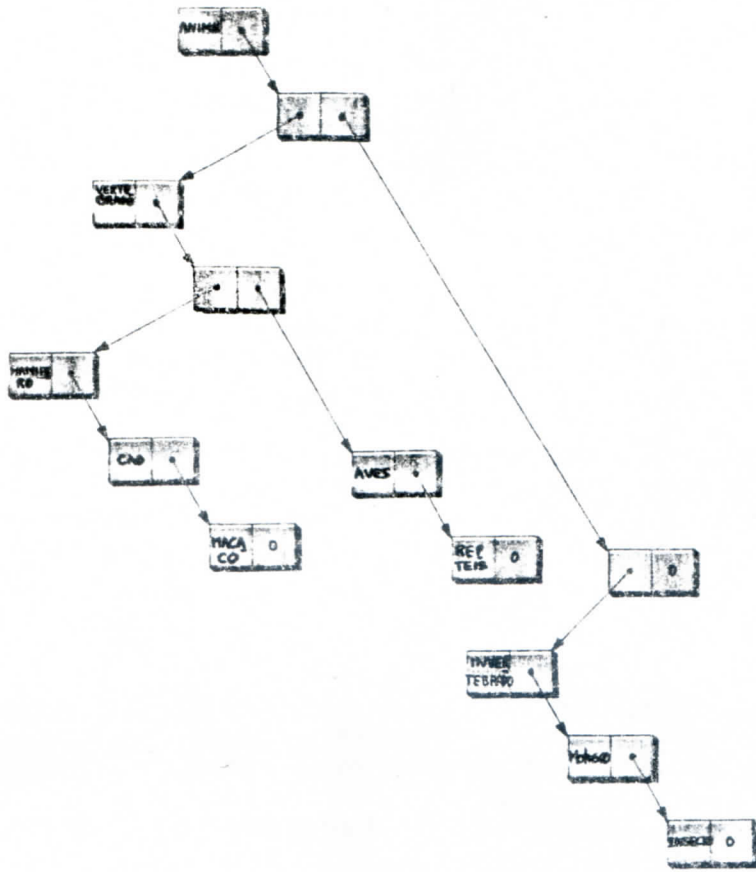
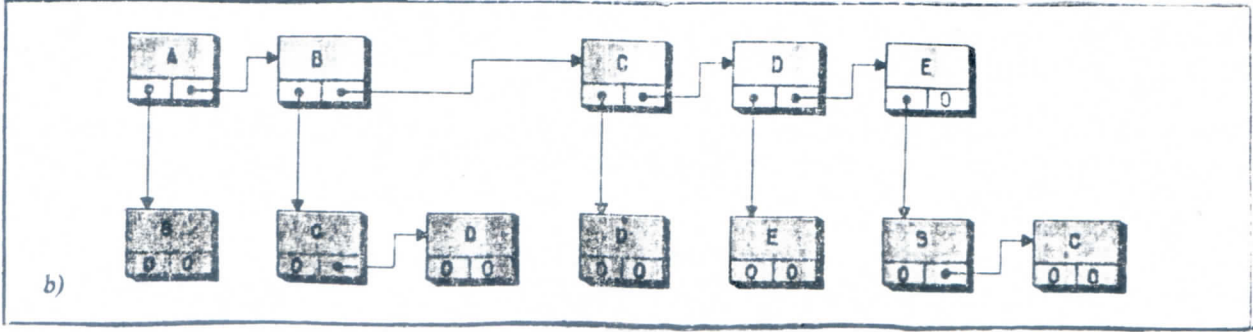
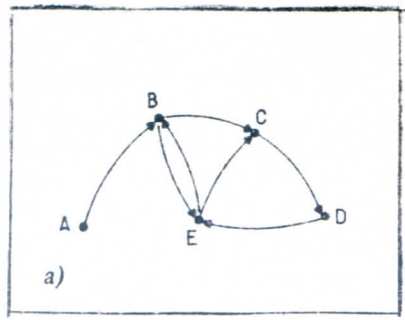


Fig. 14 - Uma arborescencia pode representar-se de varias maneiras. Esta sera utilizada nos casos em que apenas duas celulas estao disponiveis por no da arvore binaria.

Fig. 15 - Um grafico (a) sera representado sob a forma de uma arvore binaria (b), implantada por meio de quadros. As estruturas de dados constroem-se assim: estruturas cada vez mais complexas sao construidas a partir das estruturas existentes, num processo que este unicamente condicionado pela imaginacao humana.



Este artigo foi traduzido e adaptado da revista MICRO SYSTEMES, Set/Out 1982, pp.121-131.

OS EFEITOS SONOROS E O USO DOS MICROCOMPUTADORES

Todos os que possuem o modulo GERADOR DE SOM do ZX81 estarao a obter o efeito que pretendiam, quando adquiriram aquele periferico ?

Certamente que nao! O efeito obtido com essa expansao do ZX 81 e bastante diferente do efeito sonoro obtido com programas executados pelos possuidores do TI 99-4A (Texas) ou do COMMODORE 64.

Igualmente chamamos a atencao para quem planeia adquirir o BBC ou o SPECTRUM com a principal finalidade de obter efeitos sonoros. Um microcomputador que permita programar efeitos sonoros (MUSICA por exemplo), devera poder introduzir atraves do teclado, um programa, por exemplo :

```
100 TONE=110
110 FOR COUNT=1 TO 10
120 CALL SOUND(-500,TONE,1)
130 TONE=TONE+110
140 NEXT COUNT
150 END
```

este pequeno programa para o TI/99 toca os tons rapidamente

com a simples modificacao da linha 120 ... CALL SOUND(+500,TONE,1) passara a tocar 10 tons lentamente

Mas toca como ?

Em principio para que tenhamos uma audicao correcta devemos obter o efeito sonoro atraves do nosso televisor. O som e modulado e conduzido atraves do cabo da antena, podendo entao regular a altura do som com o potenciometro habitual, tal como o fazemos quando observamos um programa de TV.

Desaconselhamos efeitos sonoros atraves de altifalantes adicionais porque geralmente a qualidade nao e boa e especialmente o VOLUME e insuficiente.

No caso do SPECTRUM a modulacao em Inglaterra funciona tal como a maquina havia sido projectada, ou seja para que o som seja transmitido pelo aparelho de TV. Em Portugal, nao temos outra solucao que nao seja

```
10 FOR I=-60 TO 69
20 BEEP .2,I
30 NEXT I
```

ESTABELECEER A LIGACAO ENTRE "EAR" (do Spectrum) E "MIC" (do gravador) ATENCAO COLOQUE AS TECLAS =PLAY= e =PAUSE= ACTIVAS no seu gravador, pode agora accionar o programa com RUN

Ira escutar toda a escala musical atraves do amolificador e altifalante do seu gravador . O som nao tera um volume muito apreciavel mas sera sem duvida muito melhor do que andar a erguer o seu Spectrum ate junto do seu ouvido para escutar o efeito sonoro. Nesta amostra de programa, a instrucao BEEP e usada para obter o som, logo seguida da duracao e da frequencia, portanto quando o ciclo FOR-NEXT e executado sera percorrida a escala entre -60 e +60.

Voltamos a questao inicial, que e a seguinte : Esta a obter efeitos sonoros com o seu microcomputador ?

Nos dizemos que a melhor maquina para este tipo de realizacao e (neste momento) a da TEXAS -TI/99..... se os seus resultados sao diferentes dos nossos e se por exemplo ja conseguiu algum efeito espetacular com o MODULO DE SOM do ZX81 ou com outra maquina, entao escreva-nos a dar as suas impressoes.

```
10 BEEP .5,18: BEEP .7,14.5: G
O TO 10
```

SIRENE

```
10 FOR F=1 TO 50: BEEP .05,60-
F: NEXT F
```

ESCALA DESCENDENTE DE TONS

```
10 FOR F=1 TO 10: BEEP .01,1+F
: BEEP .01,10: NEXT F: GO TO 10
```

DISPAROS

```
10 FOR F=1 TO 30: BEEP .05,20:
BEEP .05,0: NEXT F
```

SINAL DE AVISO

```
10 FOR F=1 TO 10: FOR G=1 TO 2
: FOR H=1 TO 8: BEEP .03,24: NEX
T H: PAUSE 6: NEXT G: PAUSE 50:
NEXT F
```

TELEFONE

```
10 FOR F=1 TO 10: BEEP .5,40:
PAUSE 7: BEEP .5,40: PAUSE 35: N
EXT F
```

ALARME DE RELOGIO DIGITAL

```
10 BEEP .005,5: BEEP .005,-2:
PAUSE 2: GO TO 10
```

TRABALHAR DE UM MOTOR

```
10 FOR F=1 TO 10 STEP 2: FOR G
=1 TO 10 STEP 2: FOR H=1 TO 10 S
TEP 2: BEEP .1,F+G-H: BEEP .1,G+
H-F: BEEP .1,H+F-G: NEXT H: NEXT
G: NEXT F
```

SOM DE FUNDO
Ex:acompanhando as instrucoes
de um programa

```
10 FOR F=10 TO 20: BEEP .003,1
0: PAUSE 5-F/5: BEEP .003,5: PAU
SE 5-F/5: BEEP .003,0: PAUSE 5-F
/5: BEEP .003,1: PAUSE 25-F: NEX
T F
```

```
20 FOR F=20 TO 10 STEP -1: BEE
P .003,10: PAUSE 5-F/5: BEEP .00
3,5: PAUSE 5-F/5: BEEP .003,0: P
AUSE 25-F: NEXT F: GO TO 10
```

GALOPAR DE UM CAVALO

 *
 LOG - TECNOLOGIA INDUSTRIAL, LDA.
 AV. BOAVISTA, 832 - 2 T.
 4100 PORTO TELEF. 65127

S O F T W A R E
 =====

SINCLAIR ZX81

CONTAS CORRENTES (20, 50 E 100 CONTAS).....	1 000.00
CONTAS BANCARIAS (15 CONTAS).....	1 000.00
CAIXA (300 LANÇAMENTOS E SALDO - 16 K RAM).....	1 000\$00
SALARIOS.....	1 000\$00
ANALISE DE VENDAS (200 PRODUTOS).....	1 000\$00
RESUMO DE FACTURAS (48K E 16K RAM).....	1 000\$00
STOCKS (200 PRODUTOS - 16K RAM).....	1 000\$00
ANALISE DE INVESTIMENTOS.....	1 000\$00
PERT/CPM (ANALISE DE REDES).....	1 000\$00
ESTATISTICA.....	1 000\$00
ZX TEXTO/COMPUTACAL.....	800\$00
JOGOS (TRES CASSETES DIFERENTES).....(CADA)	500\$00
MATAMATICA I E II.....(CADA)	800\$00
XADREZ (AUSTRALIANO OU INGLES).....	600\$00
28 PROGRAMAS SINCLAIR.....	1 000\$00
ASSEMBLER/DISASSEMBLER.....	800\$00
ZX MONITOR.....	800\$00
ZX FORTH.....	1 000\$00
JOGOS LANDRY I.....	500\$00
CENTAURO.....	600\$00

 LOG - TECNOLOGIA INDUSTRIAL, LDA.
 AV. BOAVISTA, 832 - 2 T.
 4100 PORTO TELE. 65127

S O F T W A R E
 =====

SINCLAIR SPECTRUM

CONTAS CORRENTES (48K RAM).....	1	000\$00
ANALISE INVESTIMENTOS.....	1	000\$00
STOCKS (48K RAM).....	1	000\$00
VU CALC (CALCULO).....	1	000\$00
VU FILE (FICHEIROS).....	1	000\$00
VU 3D (GRAFICOS).....	1	000\$00
JOGOS I (INVADERS, LASER, LUNAR, ALIEN).....	1	000\$00
BIBLIOTECA.....		600\$00
INDICE BIBLIOGRAFICO.....		600\$00
SIMULADOR DE VOO.....		600\$00
SIMULADOR DE VOO NOCTURNO (16K).....		600\$00
XADREZ.....		600\$00
SUN-PUZZLE.....		600\$00
TRAVESSIA.....		600\$00
MASTERMIND-COMANDO-DAMAS.....		600\$00
PLANETOIDS.....		600\$00
METEOR STORM.....		600\$00
GROUND ATTACK.....		600\$00
ORBITER.....		600\$00
HORACE.....		600\$00

 LOG-TECNOLOGIA INDUSTRIAL, LDA.
 AV. BOAVISTA, 832 - 2 T.
 4100 PORTO TELEF. - 65127

S O F T W A R E
 =====

INCLAIR SPECTRUM (CONT.)

TIME GATE.....	600\$00
HORIZONS.....	600\$00
ESTATISTICA.....	800\$00
PERT/CPM.....	800\$00
MATEMATICA I (MATRIZES).....	800\$00
CALCULO DE PORTICOS (ENG. CIVIL).....	2 000\$00
SPEC-1 (CERCADO, PLANETA GROOL).....	600\$00
SPEC-2 (CAMPO DE MINAS, ATERRAGEM).....	600\$00
SPEC-3 (KINGKONG, -SALVAMENTO).....	600\$00
SPEC-4 (E.T. - 48K).....	600\$00
SPEC-5 (VINGADOR).....	600\$00
SPEC-6 (FUGA).....	600\$00

OS SOCIOS DO CLUBE Z-80 BENEFICIARAO DE-UM
 DESCONTO DE 20% SE SOLICITAREM (*) OU AD-
 QUIRIREM O SOFTWARE DIRECTAMENTE NA SEDE
 DO CLUBE

(*) EM CASO DE ENVIO PELO CORREIO, SAO
 ACRESCIDAS AS DESPESAS DE EXPEDICAO

